MAC239

Lógica de Predicados

Correção e completude

(recordação) Correção e completude da Lógica Proposicional (LP)

Teorema 1: (Correção e completude da LP).

$$\Phi_1, \ldots, \Phi_n \vdash \Psi$$

sse

$$\Phi_1, \ldots, \Phi_n \models \Psi$$

(recordação) Correção e completude da Lógica Proposicional (LP)

Teorema 1: (Correção e completude da LP).

$$\Phi_1, \ldots, \Phi_n \vdash \Psi$$

sse

$$\Phi_1, \ldots, \Phi_n \models \Psi$$

"Na LP, todo argumento que pode ser demonstrado é consequência lógica
e
toda consequência lógica pode ser demonstrada".

Correção da LPO

- Correção da LPO:

  Se $\Phi_1, \ldots, \Phi_n \vdash \Psi$ então $\Phi_1, \ldots, \Phi_n \models \Psi$

  não é surpresa: as regras da dedução natural são definidas para preservar a verdade.

Correção da LPO

- Correção da LPO:

  Se $\Phi_1, \ldots, \Phi_n \vdash \Psi$ então $\Phi_1, \ldots, \Phi_n \models \Psi$

  não é surpresa: as regras da dedução natural são definidas para preservar a verdade.

- A prova de correção da Lógica Proposicional, que faz uma indução estrutural sobre fórmulas, pode ser estendida para a LPO. Apesar das fórmulas serem mais complexas, uma vez que precisamos de argumentos sobre modelos, essa prova ainda é fácil.

Completude da LPO

- A completude da LPO

  Se $\Phi_1, \ldots, \Phi_n \models \Psi$ então $\Phi_1, \ldots, \Phi_n \vdash \Psi$

Completude da LPO

- A completude da LPO

  Se $\Phi_1, \ldots, \Phi_n \models \Psi$ então $\Phi_1, \ldots, \Phi_n \vdash \Psi$

  é mais surpreendente por que não podemos usar a prova correspondente da LP (que envolve $2^{|P|}$ avaliações com um tamanho de prova finita, apesar de eventualmente grande ...)

Completude da LPO

- A completude da LPO

  Se $\Phi_1, \ldots, \Phi_n \models \Psi$ então $\Phi_1, \ldots, \Phi_n \vdash \Psi$

  é mais surpreendente por que não podemos usar a prova correspondente da LP (que envolve $2^{|P|}$ avaliações com um tamanho de prova finita, apesar de eventualmente grande ...)

- ... mas na LPO podemos ter infinitos modelos de tipos variados! Como então provar esse teorema?

Completude da LPO

- A completude da LPO

  Se $\Phi_1, \ldots, \Phi_n \models \Psi$ então $\Phi_1, \ldots, \Phi_n \vdash \Psi$

  é mais surpreendente por que não podemos usar a prova correspondente da LP (que envolve $2^{|P|}$ avaliações com um tamanho de prova finita, apesar de eventualmente grande ...)

- ... mas na LPO podemos ter infinitos modelos de tipos variados! Como então provar esse teorema?

Terorema da Completude da LPO: *Toda consequência lógica válida pode ser demonstrada pelas regras da dedução natural da LPO*

Demonstração por indução sobre as regras de prova da LPO (tese de doutorado de Kurt Gödel em 1929)

Completude da LPO

- A completude da LPO

  Se $\Phi_1, \ldots, \Phi_n \models \Psi$ então $\Phi_1, \ldots, \Phi_n \vdash \Psi$

  é mais surpreendente por que não podemos usar a prova correspondente da LP (que envolve $2^{|P|}$ avaliações com um tamanho de prova finita, apesar de eventualmente grande ...)

- ... mas na LPO podemos ter infinitos modelos de tipos variados! Como então provar esse teorema?

Terorema da Completude da LPO: *Toda consequência lógica válida pode ser demonstrada pelas regras da dedução natural da LPO*

Demonstração por indução sobre as regras de prova da LPO (tese de doutorado de Kurt Gödel em 1929)

- Ao invés de apresentar essa prova, vamos discutir algumas de suas implicações.

Implicação do teorema da completude da LPO

Se $\Phi_1, \ldots, \Phi_n \models \Psi$ então $\Phi_1, \ldots, \Phi_n \vdash \Psi$

•A primeira consequência direta é que (como na LP) temos um jeito de mostrar que o sequente não tem uma demonstração na dedução natural: basta mostrar um modelo M que satisfaz $\Phi_1, \ldots, \Phi_n$ e não satisfaz Ψ (que corresponde à demonstração apresentando um contra-exemplo)

Implicação do teorema da completude da LPO

Se $\Phi_1, \ldots, \Phi_n \models \Psi$ então $\Phi_1, \ldots, \Phi_n \vdash \Psi$

•A primeira consequência direta é que (como na LP) temos um jeito de mostrar que o sequente não tem uma demonstração na dedução natural: basta mostrar um modelo M que satisfaz $\Phi_1, \ldots, \Phi_n$ e não satisfaz Ψ (que corresponde à demonstração apresentando um contra-exemplo)

•(recordando) **Consequência lógica da LPO**

$$\Phi_1, \Phi_2, \ldots, \Phi_n \models \Psi$$

Se $\mathcal{M} \models_\iota \Phi_i$ então $\mathcal{M} \models_\iota \Psi$

Exemplo de derivação lógica na LPO

Considere o argumento:

$$\forall x(P(x) \vee Q(x)) \vdash \forall\, xP(x) \ \vee \forall\, xQ(x)$$

Exemplo de derivação lógica na LPO

Considere o argumento:

$$\forall x(P(x) \vee Q(x)) \vdash \forall\, xP(x) \;\vee \forall\, xQ(x)$$

Vamos mostrar que ele não é um argumento válido definindo um modelo $\mathcal{M}$ que não satisfaz ambos os lados de $\vdash$:

Exemplo de derivação lógica na LPO

Considere o argumento:

$$\forall x(P(x) \vee Q(x)) \vdash \forall\, xP(x) \;\vee \forall\, xQ(x)$$

Vamos mostrar que ele não é um argumento válido definindo um modelo $\mathcal{M}$ que não satisfaz ambos os lados de $\vdash$:

Seja $A=\{a,b\}$, $\mathcal{P}^{\mathcal{M}}=\{a\}$, $\mathcal{Q}^{\mathcal{M}}=\{b\}$. Então

$$\mathcal{M} \models \forall x(P(x) \vee Q(x))$$

Exemplo de derivação lógica na LPO

Considere o argumento:

$$\forall x(P(x) \vee Q(x)) \vdash \forall\, xP(x) \;\vee \forall\, xQ(x)$$

Vamos mostrar que ele não é um argumento válido definindo um modelo $\mathcal{M}$ que não satisfaz ambos os lados de $\vdash$:

Seja $A=\{a,b\}$, $\mathcal{P}^{\mathcal{M}}=\{a\}$, $\mathcal{Q}^{\mathcal{M}}=\{b\}$. Então

$$\mathcal{M} \models \forall x(P(x) \vee Q(x))$$

mas

$$\mathcal{M} \not\models \forall xP(x) \;\vee \forall\, xQ(x)$$

Exemplo de derivação lógica na LPO

Considere o argumento:

$$\forall x(P(x) \vee Q(x)) \vdash \forall\, xP(x) \;\vee \forall\, xQ(x)$$

Vamos mostrar que ele não é um argumento válido definindo um modelo $\mathcal{M}$ que não satisfaz ambos os lados de $\vdash$:

Seja $A=\{a,b\}$, $\mathcal{P}^{\mathcal{M}}=\{a\}$, $\mathcal{Q}^{\mathcal{M}}=\{b\}$. Então

$$\mathcal{M} \models \forall x(P(x) \vee Q(x))$$

mas

$$\mathcal{M} \not\models \forall xP(x) \;\vee \forall\, xQ(x)$$

Assim, $\forall x(P(x) \vee Q(x)) \not\models \forall\, xP(x) \;\vee \forall\, xQ(x)$ e pela correção da LPO

$$\forall x(P(x) \vee Q(x)) \nvdash \forall\, xP(x) \;\vee \forall\, xQ(x)$$

Problemas de Decisão: definição

- Um *problema de decisão* é uma pergunta para algum sistema formal cuja resposta é “sim” ou “não”.
  - Responder se uma dada fórmula proposicional é SAT (UNSAT, válida e inválida) é um problema de decisão.
  - Responder se duas fórmulas são equivalentes também é um problema de decisão.
- *Decidibilidade*: problemas de decisão para os quais existe um programa que devolve “sim” quando a resposta do problema é “sim”, e devolve “não” quando a resposta do problema é “não”.

A LPO é semi-decidível

- A prova de Gödel da completude não é efetiva: ela não fornece um método para decidir se de fato um argumento é válido ou não (pode ser demonstrado ou não)
  - Na LP temos um método efetivo: temos que testar 2^n interpretações possíveis, sendo n o número de átomos na fórmula

A LPO é semi-decidível

- A prova de Gödel da completude não é efetiva: ela não fornece um método para decidir se de fato um argumento é válido ou não (pode ser demonstrado ou não)
  - Na LP temos um método efetivo: temos que testar 2^n interpretações possíveis, sendo n o número de átomos na fórmula

- Como não existe nenhuma maneira equivalente à LP para concluir que não existe uma demonstração pela dedução natural então a LPO é ***semi-decidível*** .

Decidibilidade da Lógica Proposicional

Teorema 2: *responder se uma dada fórmula da Lógica Proposicional é satisfatível, é decidível*.

Decidibilidade da Lógica de Predicados

Teorema 3: *o problema de decisão da validade de uma fórmula da LPO é* ***indecidível****: não existe um programa que, dada qualquer linguagem da lógica de predicados e qualquer fórmula Φ naquela linguagem, decida se* |= *Φ*.

Corolário:

- checar se *Φ é satisfatível é* ***indecidível.***
- checar se *Φ é um teorema é* ***indecidível***

Prova de Indecidibilidade da LPO (histórico)

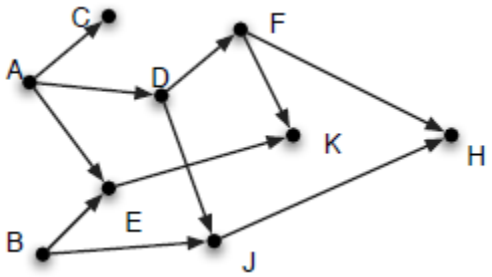
- Uma prova de indecidibilidade deve envolver um modelo formal de computação.
- O primeiro lógico a provar isso foi Alonzo Church em 1936. Seu modelo de computação foi cálculo Lambda e sua prova usa os resultados da prova de Gödel.
- Independentemente, alguns meses depois, Alan Turing criou um modelo computacional diferente e uma prova mais simples.
- As provas de Turing e Church primeiro demonstraram que não existe um algoritmo que responde perguntas sobre programas.
  - Church mostrou que é impossível decidir se 2 programas são idênticos em termos funcionais
  - Turing mostrou que é impossível decidir a parada de um programa
  - Em seguida, os dois mostraram que a LPO pode expressar as questões que eles queriam declarar sobre programas: porém *elas são indecidíveis*!
- Uma ideia geral dessa prova é dada no livro (Seção 2.5).

Expressividade da LPO (Seção 2.6)

- Muitas das propriedades de nossas especificações de sistemas computacionais e nossos programas podem ser expressos em termos lógicos.

- Naturalmente, desejamos usar linguagens lógicas mais expressivas porém, sem o custo de não ser possível raciocinar sobre elas.

Exemplo: grafo



- Um grafo dirigido G consiste de um conjunto finito de vértices V e uma relação binária entre arestas E.
- Uma maneira de representar grafos em LPO:
  - variáveis representam vértices
  - definimos o predicado binário E para a relação entre vértices. Se um grafo G contém uma aresta de u para v, então E(u,v) é verdade no modelo definido por G.
- Exemplo: a fórmula $\exists x \exists y E(x,y)$ é verdadeira para grafos que contenham pelo menos uma aresta, ou seja:

"O grafo contém pelo menos uma aresta".

Exercício (I)

Quais fórmulas expressam as seguintes declarações sobre grafos?

“O grafo tem pelo menos dois vértices”

Exercício (I)

Quais fórmulas expressam as seguintes declarações sobre grafos?

“O grafo tem pelo menos dois vértices”

$$\exists x \exists y \; E(x,y) \wedge x \neq y$$

Exercício (I)

Quais fórmulas expressam as seguintes declarações sobre grafos?

“O grafo tem pelo menos dois vértices”

$$\exists x \exists y \; E(x,y) \wedge x \neq y$$

“O grafo tem pelo menos uma aresta indo de um vértice para ele mesmo”

Exercício (I)

Quais fórmulas expressam as seguintes declarações sobre grafos?

"O grafo tem pelo menos dois vértices"

$$\exists x \exists y \; E(x,y) \wedge x \neq y$$

"O grafo tem pelo menos uma aresta indo de um vértice para ele mesmo"

$$\exists x \; E(x,x)$$

Exercício (I)

Quais fórmulas expressam as seguintes declarações sobre grafos?

"O grafo tem pelo menos dois vértices"

$$\exists x \exists y \; E(x,y) \wedge x \neq y$$

"O grafo tem pelo menos uma aresta indo de um vértice para ele mesmo"

$$\exists x \; E(x,x)$$

"O grafo contém um vértice para o qual nenhuma aresta chega"

Exercício (I)

Quais fórmulas expressam as seguintes declarações sobre grafos?

"O grafo tem pelo menos dois vértices"

$$\exists x \exists y \; E(x,y) \wedge x \neq y$$

"O grafo tem pelo menos uma aresta indo de um vértice para ele mesmo"

$$\exists x \; E(x,x)$$

"O grafo contém um vértice para o qual nenhuma aresta chega"

$$\exists x \forall y \; \neg E(y,x)$$

Exercício (II)

- “Todas as arestas do grafo são não-reflexivas”

- “Todas as arestas do grafo são bidirecionadas”

- “Todo par de vértices são conectados por um caminho de comprimento 3”

Exercício (III)

- Como expressamos (em português) as seguintes declarações formais sobre grafos?

$$\exists x \forall y E(y,x)$$

$$\forall x \forall y \, (E(x,y) \rightarrow E(y,x))$$

$$\exists x \forall y \, \neg E(x,y) \wedge \neg E(y,x)$$

Exemplo DAG

- Questão de alcançabilidade em DAGs: “*Dado um grafo G e dois nós u, v existe um caminho dirigido (sem ciclos) que vai de u a v? “*

- Essa questão é importante em várias áreas da computação: logística, *puzzles*, planejamento de caminho, etc.

- É surpreendente saber que alcançabilidade não pode ser expressa na lógica de predicados.

- Isso é provado no livro (seção 2.6)